STANFORD UNIVERSITY SCHOOL OF MEDICINE
STANFORD MEDICAL CENTER
300 PASTEUR DRIVE, PALO ALTO, CALIFORNIA

DEPARTMENT OF GENETICS

October 28, 1966

DAvenport 1-1200
Ext. 5052

Prof. François Jacob
Institut Pasteur
25, Rue Du Docteur Roux
Paris, FRANCE

Dear Prof. Jacob:

Dr. Lederberg has asked me to answer your request for a description of our experiences with computer programs for manipulation of bacterial stock collection data.

Early in 1963, I wrote such a program for our B. subtilis collection. The program provided for most of the obvious interrogation demands one might make on a bacterial strain library, e.g. pedigree trees and progeny lists for given strains, strain lists for given marker subsets, etc., and was so designed that its powers could be extended, if desired, by the addition of new subroutines. Although the program was written in a language (SUBALGOL) indigenous to the Stanford University IBM 7090 system, a translated (compiled) version is available as a FORTRAN Assembly Program (FAP) card deck, a form acceptable to most other IBM 7090 or 7094 systems.

However, the 1963 program was intended primarily as a prototype, an excercise to see what could be done, what problems would arise, and what changes, if any, in the actual organization of our strain data might be worth the trouble. For a variety of reasons, I definitely do not recommend current use of that program, but advise instead a fresh start, building on the 1963 experience as background. Below, I will set down some general considerations arising from that experience, a brief outline of the 1963 program to highlight its shortcomings, and some retrospective comments, all of which I hope may be of some help to you in formulating your own approach to the matter.

The central problem is, quite obviously, to devise a representation of the strain, the associated information (origin, genotype, etc.), and the collection as a whole which permits efficient storage, manipulation, and retrieval of the data. Efficiency here applies both to space and time. Depending on the mnemonic medium used (core, disc, tape, etc.), these two aspects of efficiency may conflict and enforce a compromise. The solution of the representation problem and the choice of storage medium will in part be tailored to fit the immediate retrieval demands that are to be made on the system, but one should avoid making the fit too close and should attempt to build in a measure of flexibility which will permit some ease of response to queries which may not at present seem relevant. A corollary of such flexibility is that the master program itself should be written in such a way that additions to it can be easily made.

Decisions concerning the format and medium of input-output are largely peripheral to solution of the central problem. Computer efficiency during actual search and arrangement operations on the central data bank will be unaffected by the physical and representational form assumed by data at input and output. However, serious losses of overall efficiency can result if the translation steps between external and internal representation are awkward and time-consuming. To this extent, the central storage and input-output problems do overlap, and some compromises may have to be made on either side; but, more often that not, one can exploit cleverness in programming the translation steps themselves to relieve the strain.

The proper motivation for decisions about input=output should arise from the fact that these areas define the interface between man and computer, and, whenever possible, the man should be favored in the interaction. Thus, convenience, simplicity, and clarity should be the overriding concerns on designing the input=-output facilities.

Briefly, the approach of the 1963 program to the general goals described above was as follows:

During "run-time" (actual computation), the entire B. subtilis collection was put into core storage (which provides by far the fastest access time of available forms of memory) along with the program itself. This was possible because the number of subtilis strains was small (<103), but even so, the limited size of the core bank (3.2×10^4 words of 36 bits each in the IBM 7090) demanded some sophistication in data packing techniques with a consequent slight, but not significant, sacrifice in access time. Input was initially (the first time the program was run) a punched card deck containing the SUBALGOL program followed by the strain collection in a representation on cards not greatly different from that employed in the manually kept stock book. Some minor alterations were made in the designation of markers. An outline of the general considerations underlying these changes was presented to Demerec's group on genetic nomenclature which met during the Cold Spring Harbor Symposium of 1963. A copy of that outline is enclosed. Subsequent inputs employed a FAP deck produced by the initial run and a few cards containing retrieval requests and additions to the collection, if any. The FAP deck was essentially a punched-card image of the core bank status at the time the deck was made and therefore included both the program, in machine language, and the stock collection, in machine code. To summarize, storage during computation was on core, interim storage was on punched cards used also as input. Output was predominantly in printed form with formats varying to most clearly present the kind of data requested; some specialized output was in the form of punched cards.

Though having to guess somewhat at the scope of the project you have in mind and the computer facilities at your disposal, I can make the following comments and suggestions relating our experience with the solutions of 1963 to the problems facing you:

First, your collection must be larger, by an order of magnitude or more, than 10^3 strains, and the genetic information for each strain is probably more extensive than in the subtilis case. Thus, given the modest size of core banks in even the newest computers, it will probably not be possible to read all your data into core at once. Some hybrid system will be necessary, preferably with tape or disc as the auxiliary depot from which segments of the collection are called into core for use by the program. This is not a very great hardship since many routine demands on libraries involve only segments of the collection anyway. To make such segments as large as possible, some effort at efficiency of data packing will probably be advisable. It may even be worthwhile to consider economies which may be won from revisions in the general methods of genotypic definition. For example, to avoid long lists of markers associated with every strain, one might enlarge the category of "reference strains" beyond the usual wild or source types in a way which would optimally condence the description of the majority of strains. For very large collections, optomization of such revisions and the bookkeeping of the revisions themselves are problems best handled by "editor" programs. Some form of "editor", or even a series of them, is almost indispensable for the initial stages in development of a storage system for an existing data set.

Second, toward the end of efficiency in data packing and translation, I made use of some tricks based on idiosyncrasies of the IBM 7090 computer (e.g. the internal binary code representation of alphanumeric characters) and thus rendered the 1963 program "machine dependent," that is to say useless (without extensive alteration) to any location with a computer other than the IBM 7090 or 7094, and thus useless even here at Stanford when the 7090 is retired. This was a mistake which I hope never to repeat and one which I warn you against making even once. The rate of advance in computer engineering continues to shorten the practical lifetime of individual designs, and prudence dictates that any program which is to be used over a number of years be written, in so far as foresight permits, so as to require the fewest and simplest alterations when the new model replaces the computer you are using now. The seriousness of this problem is acknowledged by computer scientists and programmers, but the most important single step toward solving it, the adoption of an international standard programming language recognized by the computer industry as well as by computer users, has not yet been taken. However, continuing efforts are being made to promote ALGOL (see Introduction to ALGOL, Baumann, et.al., Prentice-Hall, Inc., 1964) as such a language, and one might be on fairly safe ground if the program were written in it or a dialect of it and if no tricks were employed that depended heavily on the actual computer system being used.*

Last, I would advise against the use of punched cards for either interm storage or input and interrogation, that is, for any purpose whatsoever. They wear out easily and must be reproduced almost as frequently as they are used. In large numbers, they become so awkward and unwieldy as to discourage the casual user from carrying them to the computer and thus virtually destroy one of the major attractions (ready accessibility) of a computer managed library. At Stanford in 1963, punched cards were the only generally available means of communication between the large computers and the community of users, and we were forced to use them in spite of their disadvantages. Clearly cards marked a transient phase in the use of computers; here at Stanford they are already being replaced by teletypewriter substations of large time-sharing systems. Interim storage of the collection should be on tape, disc, or drum, and input, retrieval requests, and output should be by teletype to invite frequent use. If cards must be employed at present, the program should be designed to facilitate transfer to the more satisfactory media as soon as they become available.

Please feel free to request clarification, if any of the above discussion seems turbid, or answers, if you think I may have them, to specific questions that arise.

Sincerely,

Larry Okun,
Department of Genetics

*Dr. Lederberg remarks that a "list-processing language" like LISP would have many built-in facilities of great use for file searches of genetically connected data.